

SYSTEM AND METHOD FOR TRANSPARENT STORAGE REORGANIZATION

FIELD OF THE INVENTION

5 The invention relates generally to computer systems, and more particularly to an improved system and method for transparent storage reorganization.

BACKGROUND OF THE INVENTION

10 Storage reorganization may be motivated by any number of reasons. For example, storage migration and consolidation are important operations that help reduce the total cost of ownership of storage servers in an enterprise. One important factor in reducing total cost of ownership is to reduce the
15 time and expense required to manage storage servers, and thus the number of servers. Unsurprisingly, storage consolidation often occurs in conjunction with an upgrade to a newer, more performant server version such as moving from Microsoft® Windows NT Server version 4.0 to Microsoft® Windows Server
20 2003. Storage administrators may take advantage of such an upgrade to reduce management overhead by consolidating storage from many legacy servers onto one or fewer new machines that may serve all the content that was on the legacy servers.

 In relocating file shares from two servers, such as "foo"
25 and "bar", one common strategy deployed is to map both the

server name "foo" and the server name "bar" to the same storage server. One problem that may occur in this reorganization of storage is a resulting name clash from trying to relocate two file shares with the same name, such as
5 \\foo\public and \\bar\public, onto the same storage server.

Normally two shares would need to be created, both with the path name \\server\public. Such a name clash could be avoided by renaming one or both of the relocated shares, using one or more well known techniques such as versioning. However,
10 system administrators may be reluctant to consolidate legacy shares if they must modify the file share names to avoid such a name clash.

Modifying a file share name visible to a user to avoid a name clash raises several problems. First of all, modifying a
15 file share name visible to a user during reorganizing storage would require training users to use the new names.

Furthermore, file share path names that are embedded in documents, web pages, and applications would need to be located and the old names would need to be changed to the new
20 names. These burdensome activities would require additional time and expense for a storage administrator managing reorganization of storage.

What is needed is a way for a storage administrator to reorganize storage using legacy share names so that users or

clients may access the relocated legacy shares using the legacy share names. Any such system and method should allow a system administrator to easily and efficiently monitor client access to the relocated legacy shares so that the storage administrator may retire relocated legacy shares that are infrequently used.

SUMMARY OF THE INVENTION

Briefly, the present invention provides an improved system and method for transparently reorganizing storage. To this end, a reorganization server is provided that may relocate legacy shares from one or more legacy servers onto one or more other servers. The reorganization server may include a distributed file system with a path rewriter for transparently prepending another server name to the received legacy path, and with a path redirector for resolving any links to a new storage location that may be encountered while traversing a rewritten legacy share path. The reorganization server may also include a database or log file for recording access and usage information of relocated legacy shares.

The present invention may transparently reorganize storage by first aliasing the name of a legacy server to the network address of a reorganization server. The contents and permissions of each legacy share may then be copied to a

unique share name on another server. A root may next be created on the reorganization server using the legacy server name, and a link that points to the legacy share copied on the other server may be created on that root. Any client may then
5 request access to the relocated legacy share using the legacy share name. The reorganization server may rewrite the received legacy path by prepending it with the reorganization server name, resolve any links in the rewritten legacy share path name, and respond with the share path name of the storage
10 location of the relocated legacy share.

Advantageously, the system and method may be used to re-architect and integrate previously disparate file shares into a single namespace. Moreover, the present invention may be used to expand storage from a few servers to many servers in
15 addition to consolidating storage from many servers to fewer servers. By recording information about access to and usage of the legacy shares on a single reorganization server, a storage administrator may easily monitor access and usage of legacy shares for management and archival purposes.

20 Furthermore, the system and method provided are flexible and extensible so that any file system or name resolution protocol equipped with path rewriting and path redirection capabilities may be used. Thus, the redirection provided by the present invention may be broadly supported across share path names,

server names, file system protocols, and other data access protocols.

Other advantages will become apparent from the following detailed description when taken in conjunction with the
5 drawings, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram generally representing a computer system into which the present invention may be
10 incorporated;

FIG. 2 is a block diagram generally representing an exemplary architecture of system components for reorganizing storage, in accordance with an aspect of the present invention;

15 FIG. 3 is a flowchart generally representing the steps undertaken for reorganizing shares from a legacy server onto another server, in accordance with an aspect of the present invention;

FIG. 4 is an exemplary illustration generally
20 representing a consolidation server consolidating shares from a legacy server onto a new server, in accordance with an aspect of the present invention;

FIG. 5 is a flowchart generally representing the steps undertaken for accessing a reorganized share moved from a

legacy server onto another server, in accordance with an aspect of the present invention;

FIG. 6 is an exemplary illustration generally representing a client accessing a legacy share that has been consolidated on another server, in accordance with an aspect of the present invention;

FIG. 7 is an exemplary illustration generally representing a consolidation server performing name redirection for consolidated legacy shares placed on several other servers, in accordance with an aspect of the present invention; and

FIG. 8 is an exemplary illustration generally representing a consolidation server hosting a subset of consolidated legacy shares, in accordance with an aspect of the present invention.

DETAILED DESCRIPTION

EXEMPLARY OPERATING ENVIRONMENT

FIG. 1 illustrates an example of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing

environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to:

personal computers, server computers, hand-held or laptop devices, tablet devices, headless servers, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, and so forth, which perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in local

and/or remote computer storage media including memory storage devices.

With reference to FIG. 1, an exemplary system for implementing the invention includes a general purpose
5 computing device in the form of a computer 110. Components of the computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be
10 any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus,
15 Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

The computer 110 typically includes a variety of computer-readable media. Computer-readable media can be any
20 available media that can be accessed by the computer 110 and includes both volatile and nonvolatile media, and removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media

includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer.

110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer-readable media.

The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic
5 routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of
10 example, and not limitation, FIG. 1 illustrates operating system 134, application programs 135, other program modules 136 and program data 137.

The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By
15 way of example only, FIG. 1 illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a
20 removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards,

digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic
5 disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

The drives and their associated computer storage media, discussed above and illustrated in FIG. 1, provide storage of
10 computer-readable instructions, data structures, program modules and other data for the computer 110. In FIG. 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146 and program data 147. Note that these components
15 can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers herein to illustrate that, at a minimum,
20 they are different copies. A user may enter commands and information into the computer 110 through input devices such as a tablet, or electronic digitizer, 164, a microphone 163, a keyboard 162 and pointing device 161, commonly referred to as mouse, trackball or touch pad. Other input devices not shown

in FIG. 1 may include a joystick, game pad, satellite dish, scanner, or other devices including a device that contains a biometric sensor, environmental sensor, position sensor, or other type of sensor. These and other input devices are often
5 connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected
10 to the system bus 121 via an interface, such as a video interface 190. The monitor 191 may also be integrated with a touch-screen panel or the like. Note that the monitor and/or touch screen panel can be physically coupled to a housing in which the computing device 110 is incorporated, such as in a
15 tablet-type personal computer. In addition, computers such as the computing device 110 may also include other peripheral output devices such as speakers 195 and printer 196, which may be connected through an output peripheral interface 194 or the like.

20 The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes

many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160 or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

TRANSPARENT STORAGE REORGANIZATION

The present invention is generally directed towards a method and system for providing transparent storage
5 reorganization. As used herein, storage reorganization means any relocation of file shares from one computer to another computer. A share means, as defined herein, a set of files and directories exposed via a remote file system or data access protocol. A legacy share means a previously existing
10 share, such as shares used on servers in various businesses, governments, and other organizations.

One form of storage reorganization is storage consolidation. Storage consolidation means herein any relocation of file shares to one or more other storage
15 systems, including storage servers, that may result in a reduction of the set of storage systems. As will be understood, the various block diagrams, flow charts and scenarios described herein are only examples, and there are many other scenarios to which the present invention will
20 apply.

Storage may be reorganized in many ways using different system configurations. As an example, the system of FIG. 2 will be described using one embodiment of a system for reorganizing storage, a consolidation server. A consolidation

server may reorganize storage by consolidating storage from one or more legacy servers. As will be understood, the system for consolidating storage from one or more legacy servers is one example of many system configurations that may use the
5 system components described for reorganizing storage.

Turning to FIG. 2 of the drawings, there is shown a block diagram generally representing an exemplary architecture of system components for reorganizing storage. Those skilled in the art will appreciate that the functionality implemented
10 within the blocks illustrated in the diagram may be implemented as separate components or the functionality of several or all of the blocks may be implemented within a single component. For example, the functionality for the database 210 may be included in the distributed file system
15 204. Or the functionality of the path rewriter 206 may be may be implemented as a separate component.

The server 202 may include a distributed file system 204 and a database 210. In general, the distributed file system 204 and the database 210 may be any type of executable
20 software code such as a kernel component, an application program, a linked library, an object, and so forth. The distributed file system 204 may include an operably coupled path rewriter 206 and an operably coupled path redirector 208. Each of these components may also be any type of executable

software code such as a kernel component, an application program, a linked library, an object, or other type of executable software code.

In specific, the distributed file system 204 may be any
5 file system with path rewriting and path redirection implemented as shown in FIG. 2 by the path rewriter 206 and the path redirector 208, respectively. Such a file system may receive requests for accessing a file share using any type of file system protocol, including the Microsoft® Distributed
10 File System (Dfs) using the Server Message Block (SMB) protocol for accessing file shares or using other types of file system protocols that provide similar characteristics, such as the NetBIOS protocol, the Network File System protocol (NFS), the Netware File Sharing Protocol (NFSP), or another
15 protocol. In one embodiment, a Dfs service augmented with path rewriting and path redirection may be run on the server 202. This Dfs service may be enabled, for instance, through an operating system configuration setting such as a registry key in the registry of the operating system of the server 202.

20 The distributed file system 204 may receive a request to access a file. The path rewriter 206 may automatically rewrite any path for a legacy share to prepend another server name to the legacy server name. For example, in an embodiment such as the consolidation server, the path rewriter 206 may

rewrite the legacy share path to prepend the consolidation server name to the legacy share path. After rewriting the path name, the distributed file system 204 on the consolidation server may continue normal processing on the rewritten path name. As part of its normal processing, the distributed file system 204 may access the Dfs root corresponding to the legacy server name, traverse the legacy share path name, and encounter a link pointing to a storage location of a relocated legacy share. In this case, the distributed file system 204 may invoke the path redirector 208.

The path redirector 208 may resolve any links encountered while traversing a legacy share path, including a rewritten legacy share path. Such a link may point to any kind of path that may be supported by file system protocols, such as Dfs, SMB, NetBIOS, NFS, NFSP or other type of protocol. Upon resolving the link, the distributed file system may respond with the share path of the storage location of the relocated legacy share.

In one embodiment of the consolidation server, the consolidated storage may be organized in a separate Dfs namespace. In this case, the link may contain the path in the namespace that corresponds to the consolidated legacy share. Advantageously the path redirector 208 of the consolidated

server may, in turn, redirect that path to the new namespace and later the storage may be moved or reorganized within the new Dfs namespace without impacting the redirection into the new Dfs namespace.

5 A database 210 is operably coupled to the distributed file system 204 so that access activity and usage of legacy shares may be recorded and monitored. The database may be any type of database or may be a log file. Whenever the server sees a request for a legacy share, the distributed file system
10 204 may log information about the request into the database or log file. Any information about access to the legacy share may be recorded in the database, such as the name of legacy share that was accessed, which client requested access to the share, etc. This information may be used by a storage
15 administrator in a number of ways. For instance, a storage administrator may track the active usage of the legacy shares. If a storage administrator decides that the shares are infrequently used, then the administrator may retire them. As another example, the storage administrator may use the
20 information to map which user or applications are accessing the legacy paths so that the storage administrator may decide what applications need to be updated with the new path name, or alternatively, to decide which users to notify to update their links to the new path name.

The system described for consolidating storage from one or more legacy servers is one example of many system configurations that may use the system components shown in FIG. 2 for reorganizing storage. Other system configurations for reorganizing storage may include a distribution server that may use the system components described for replacing a monolithic server by dispersing legacy file shares from the monolithic server across one or more smaller servers. Yet another system configuration may include a transfer server that may use the system components described for replacing a legacy server with a replacement server by transferring legacy file shares on the legacy server to the replacement server.

FIG. 3 presents a flowchart generally representing the steps undertaken for reorganizing shares from a legacy server onto another server. At step 302, any legacy server name may be aliased to the network address of a reorganization server. The legacy shares may be stored on the reorganization server or on another destination server. Aliasing the legacy server name may result in name lookups on the legacy server names resolving to the reorganization server so that any requests for the legacy server are instead made to the reorganization server for handling the request. This aliasing may be performed for all naming schemes used for the legacy server, such as Domain Name System (DNS) and NetBIOS names, to ensure

that any lookups for the legacy server name through any protocol will actually result in the address of the reorganization server where the request will be handled.

After aliasing the legacy server name with the
5 reorganization server name, the contents and permissions of each legacy share may be copied to a destination server at step 304. In one embodiment, the reorganization server may also serve as a destination server for one or more legacy shares. At step 306, each legacy share is assigned a new
10 unique share name. In one embodiment, this unique share name may not be visible to a user or client machine. Using the legacy server name, a legacy server Dfs root may be created on the reorganization machine at step 308. In one embodiment, the legacy server Dfs root may not have the identical name of
15 the legacy server; instead, the legacy server Dfs root name may be a transformation of the legacy server name, such as prepending an identifier string to the legacy server name. At step 310, a Dfs link may be created on the legacy server root to the share name on the destination server where the legacy
20 share was copied. After the link is created on the legacy server root, the copied legacy shares may be accessed from the destination server. Those skilled in the art will appreciate that the steps described in FIG. 3 may be performed in a different order for reorganizing shares from a legacy server

onto another server. For example, a legacy server root and a link on the legacy server root may be created on the reorganization machine before the contents and permissions of the legacy files are copied to a destination server.

5 In one embodiment of the present invention, server 202 may be a consolidation server that may consolidate legacy shares from legacy servers onto a smaller set of destination servers. FIG. 4 presents an exemplary illustration generally representing a consolidation server consolidating shares from
10 a legacy server onto a new server. In this embodiment, there may be a number of legacy servers, M_1 to M_i , such as legacy server M_i 402. Each of these servers may have a file system 404 with one or more shares, S_1 to S_j , accessed using a path name such as $\\M_i\\S_j$. A storage administrator may want to
15 consolidate these shares on a smaller set of new storage servers, such as the new server NS, so that client machines may continue accessing any share by using the same path name.

 However, due to popularity of some share names, a share located on one machine may have the same name as a share
20 located on another machine. For example, the name "public" is a common share name. There may be a share name of "public" on machines M_i and M_k that may be accessed using the path names $\\M_i\\public$ and $\\M_k\\public$, respectively. When two shares with the same name are consolidated onto the same server, a

name clash may occur unless one or both shares are assigned a unique path name for access. Typically, the path name for accessing one or both shares is changed to avoid the name clash. However, the present invention may allow a storage administrator to consolidate these shares on a smaller set of storage devices so that client machines may continue accessing any share by using the same path name, even in the event of a name clash. To do so, the consolidation server CS 406 may include the distributed file system 204 with path rewriting implemented by the path rewriter 206 and with path redirection implemented by the path redirector 208 as described in FIG. 2.

The first step in FIG. 4 is to alias the legacy server name M_i to the network address of the consolidation server CS 406 to cause the name of the legacy server, in this case M_i , to resolve to the network address of the consolidation server CS, so that any requests for the legacy server M_i are instead directed to the consolidation server CS. This aliasing may be performed for all naming schemes used for the legacy server, such as Domain Name System (DNS) and NetBIOS names, to ensure that any lookups for the legacy server name through any protocol will actually result in the address of the consolidation server CS rather than that of the legacy server, M_i .

The second step may be performed for every consolidated legacy share. Each legacy share S_j to be consolidated may be copied from the legacy server M_i 402 to the new server NS 410. The contents and permissions of the share S_j may be copied to
5 the new server NS 410 and the share S_j may be given a new unique share name that may not be visible to a user or client machine. Any name may be used for the new share that does not cause a name conflict. In one embodiment, the naming scheme "\\NS\\ M_i - S_j " may ensure that no such conflicts occur.

10 The third step may be performed once per consolidated legacy server M_i . Using the legacy server name, a legacy server root, such as a new DFS root with the name M_i , may be created on the consolidation server CS 406. As a result, the consolidation server CS may respond, or handle locally,
15 accesses to paths of the form "\\CS\\ M_i ". With the path rewriter 206 enabled, whenever the distributed file system 204 of the consolidated server CS 406 receives a path name beginning with the server name of M_i , such as "\\ M_i ", it may rewrite the path as "\\CS\\ M_i " followed by the rest of the path. Thus, the
20 distributed file system 204 may find that the rewritten path corresponds to a local Dfs root and then may access that local root according to the Dfs protocol.

Note that, in one embodiment, a character that is illegal for the first character of a root or share name, such as a

hash mark, may be prepended to the legacy server name before using it to create a legacy server root on the consolidation server. By so modifying the legacy server name to create a legacy server root, there may be guaranteed that no clash will occur between a local share name on the consolidation server CS and a root name of a legacy server. Assuming no local root or share name may begin with such a character, a name for a local share on the consolidation server CS and a name from a legacy server for a consolidated share may be quickly identified. This may advantageously allow a storage administrator to quickly identify which roots on the consolidation server CS are local roots and which roots are from consolidated shares.

The fourth step may be performed once per legacy share S_j that is consolidated on the new server. A link may be created on the legacy server root $\\CS\\M_i$ of the consolidation server CS to the share name $\\NS\\M_i-S_j$ on the new server NS 410 where the legacy share was copied. Such a link may point to any kind of path that may be supported by the distributed file system protocols, including Dfs, SMB, NetBIOS, NFS, NFSP or other type of protocol. In one embodiment, the link may be a Dfs link that points directly to the location of the consolidated legacy share on the new server NS 410. In this case, a path beginning with $\\CS\\M_i\\S_j$ may be traversed from

the root \\CS\M_i until the link S_j is encountered and redirects the request to the new location of the consolidated legacy share \\NS\M_i-S_j of the file system 412 on the new server NS 410. In another embodiment of the consolidation server, the consolidated storage may be organized in a separate Dfs namespace. In this case, the link may contain the path in the namespace that corresponds to the consolidated legacy share. Advantageously the path redirector 208 of the consolidated server may, in turn, redirect that path to the new namespace and later the storage may be moved or reorganized within the new Dfs namespace without impacting the redirection into the new Dfs namespace and without requiring any configuration changes on the consolidation server CS.

After the link is created on the legacy server root, the consolidated legacy shares are now accessible on the new server NS 410. Those skilled in the art will appreciate that the steps described in FIG. 4 may be performed in a different order for consolidating shares from a legacy server onto a new server. For instance, a legacy server root and a link on the legacy server root may be created on the consolidation server CS 406 before the contents and permissions of the legacy files are copied to the new server NS 410. After the legacy shares have been consolidated onto a new server, the legacy server may then be retired or renamed and used for other purposes.

FIG. 5 presents a flowchart generally representing the steps undertaken for accessing a reorganized share moved from a legacy server onto another server. A client may wish to access a reorganized legacy share using the legacy share name.

5 At step 502, the client may resolve the aliased legacy server name and establish a connection to the network address of the reorganization server CS. The client may then send a request to the reorganization server to access the legacy share at step 504. In one embodiment, the request may be a Dfs request
10 for establishing access to the legacy share.

At step 506, the reorganization server may rewrite the legacy share path by prepending the the reorganization server name to the legacy share path so that the distributed file system may find that the rewritten path corresponds to a local
15 root with the legacy server name and then may access that local root. After rewriting the legacy share path, the distributed file system on the reorganization server continues normal processing on the rewritten path. As part of its normal processing, the distributed file system 204 may
20 traverse the rewritten legacy share path and encounter a link pointing to a storage location of a relocated legacy share. In this case, the distributed file system 204 may resolve any links encountered while traversing the rewritten legacy share path at step 508 by invoking the path redirector 208. Such a

link may point to any kind of path that may be supported by the distributed file system protocols, including Dfs, SMB, NetBIOS, NFS, NFSP or other type of protocol.

Upon resolving the link, the distributed file system may
5 respond at step 510 with the share path of the storage location of the relocated legacy share. The client may, in turn, access the legacy share at step 512. Information about the redirection to the legacy share may also be recorded in the database. In one embodiment of the reorganization server,
10 the relocated storage may be organized in a separate Dfs namespace. In this case, the link may contain the path in the namespace that corresponds to the reorganized legacy share. Advantageously the path redirector 208 of the reorganization server may, in turn, redirect that path to the new namespace
15 and later the storage may be moved or reorganized within the new Dfs namespace without impacting the redirection into the new Dfs namespace.

Returning to the embodiment of the present invention where the reorganization server may be a consolidation server,
20 FIG. 6 presents an exemplary illustration generally representing a client accessing a legacy share that has been consolidated on another server as previously described and shown in FIG. 4. In this embodiment, the client 602 may send an access request to the consolidation server CS 406 for the

legacy share `\\Mi\Sj`. In the first step of FIG. 6, the client who wants access to a legacy path `\\Mi\Sj` may resolve the server name `Mi` that is aliased to the consolidation server CS 406 either through DNS or NetBIOS, or any other name resolution protocol. The client may then connect to the network address of the consolidation server CS 406 via the SMB protocol. The client and the server may negotiate that the legacy share is a Dfs share and the client may then send a create request to the consolidation server CS 406 for the legacy path `\\Mi\Sj`.

In the second step, the consolidation server executing the distributed file system 204 with the path rewriter 206 enabled may receive the create request for the legacy path `\\Mi\Sj`. When any type of request for a share is received with the path rewriter enabled, the distributed file system on the consolidation server may automatically rewrite any path for a legacy share by prepending the consolidation server name to the path. For example, upon receipt of a request for a share with the path name of `\\Mi\Sj`, the distributed file system on the consolidation server may automatically rewrite this path to `\\CS\Mi\Sj` before performing any processing on the path name. As a result, the consolidation server CS may respond, or handle locally, accesses to paths of the form `\\CS\Mi`. After rewriting the path name, the distributed file system on

the consolidation server may continue normal processing on the rewritten path name. The distributed file system 204 may find that this path corresponds to a local root and then may access the local root M_i . In one embodiment, if the consolidation
5 server cannot find any root using the rewritten path
 $\\CS\\M_i\\S_j$, it may automatically revert back to the original path $\\M_i\\S_j$ sent by the client and may assume that this is a local root that was not consolidated.

In step three, the distributed file system may traverse
10 the legacy share path name and encounter a link pointing to a storage location of a relocated legacy share, such as the link S_j in the legacy path $\\M_i\\S_j$ of the file system 408. In one embodiment, an SMB server on the consolidation server may traverse the legacy share path and discover a reparse point
15 that indicates S_j is a link and may return a message, such as STATUS_PATH_NOT_COVERED, indicating to the client 602 that S_j is a link. In step four, the client 602 may send a referral message for $\\M_i\\S_j$ to the consolidation server. Upon
receiving the message, the consolidation server CS may then
20 rewrite the referral request path $\\M_i\\S_j$ to $\\CS\\M_i\\S_j$ in step five and may determine using the path redirector 208 that the rewritten path $\\CS\\M_i\\S_j$ maps to a link to $\\NS\\M_i-S_j$. Note that there may be many links within a path and this process may be repeated, on the same or on different servers, for each

link discovered in the path. In one embodiment, if the consolidation server cannot find any root using the rewritten path \\CS\M_i\S_j, it may automatically revert back to the original path \\M_i\S_j sent by the client and may assume that
5 this is a local root that was not consolidated.

In step six, the consolidation server responds to the client with a referral to the share path \\NS\M_i-S_j which is the new location of the consolidated legacy share. The client may then access the path \\NS\M_i-S_j in step seven. In one
10 embodiment, the client machine may automatically cache the referral to the location of the consolidated legacy share so that the client may directly access the location of the consolidated legacy share for any additional access, or for a specified period of time, without needing to go through the
15 consolidation server or require redirection.

Although FIGs. 4 and 6 show one consolidation server, there may be one or more consolidation servers that may perform name redirection for the consolidated legacy shares. Moreover, in one embodiment, the legacy shares may be placed
20 on one or more other servers as illustrated in FIG. 7. The file system 408 of consolidation server 406 in FIG. 7 may include a local root for legacy server names M₁ to M_N, each of which root has a link S_j to the share name on a new server where the legacy shares for that legacy server are copied.

For instance, the file system 408 of consolidation server 406 includes the local root for legacy server name M_1 with a link S_j that points to the consolidated legacy shares $\\NS_1\\M_1-S_j$ on file system 704 of New Server 1 702. Similarly, the file system 408 of consolidation server 406 includes the local root for legacy server name M_N with a link S_j that points to the consolidated legacy shares $\\NS_N\\M_N-S_j$ on file system 712 of New Server N 710.

In addition to performing name redirection for the consolidated legacy shares, the consolidation server may itself host a subset of the consolidated legacy shares in another embodiment as illustrated in FIG. 8. The file system 408 of consolidation server 406 in FIG. 8 may include a local root for legacy server names M_1 to M_N . With the exception of the local roots for legacy server names M_2 and M_3 , each of the local roots has a link S_j to the share name on a new server where the legacy shares for that legacy server are copied. For these consolidated legacy share, the consolidation server 406 may perform name redirection. For instance, the file system 408 of consolidation server 406 includes the local root for legacy server name M_4 with a link S_j that points to the consolidated legacy shares $\\NS_2\\M_4-S_j$ on file system 708 of New Server 2 706. However, the file system 408 of consolidation server 406 may itself host the consolidated

legacy shares $\backslash M_2 \backslash S_j$ and $\backslash M_3 \backslash S_j$, under the share names
 $\backslash CS \backslash M_2 - S_j$ and $\backslash CS \backslash M_3 - S_j$, respectively. The consolidation
server 406 may as well perform name redirection for these
consolidated legacy shares. Those skilled in the art will
5 appreciate that in another embodiment, the consolidation
server alone could host all of the consolidated shares.

Not only may the system and method transparently
reorganize storage, the present invention may advantageously
allow for monitoring access and usage of reorganized legacy
10 shares at a single location rather than multiple locations in
a distributed file system. Whenever the consolidation server
sees a request for a legacy share, the distributed file system
204 may log information about the request into the database or
log file. Any information about access to a legacy share may
15 be recorded in the database, such as the name of the legacy
share that was accessed, which client requested access to the
share, etc. This information may be used by a storage
administrator to track the active usage of the legacy shares
and retire any shares infrequently used. Moreover, a storage
20 administrator may use the information to map which user or
applications are accessing the legacy paths so that the
storage administrator may decide what applications need to be
updated with the new path name, or alternatively, to decide

which users to notify to update their links to the new path name.

As can be seen from the foregoing detailed description, the present invention provides an improved system and method
5 for transparently reorganizing storage so that a client or user may access the reorganized storage using the legacy share name. Advantageously, legacy names embedded in documents, web pages, and applications do not need to be changed to the path name for the new storage location of a relocated legacy share,
10 nor do users need to be trained to use the relocated legacy share path name. As is now understood, the system and method described for consolidating storage from one or more legacy servers is one example of many system configurations that may use the present invention for reorganizing storage. Other
15 system configurations for reorganizing storage may include a distribution server that may use the present invention for replacing a monolithic server by dispersing legacy file shares from the monolithic server across one or more smaller servers. Thus the present invention may be used to expand storage from
20 a few servers to many servers as well as to consolidate storage from many servers to fewer servers. Yet another system configuration may include a transfer server that may use the system and method for replacing a legacy server with a replacement server by transferring legacy file shares on the

legacy server to the replacement server. Furthermore, the system and method provided are flexible and extensible so that any file system or data access system using a file system or name resolution protocol with path rewriting and path redirection implemented may be used. Moreover, the redirection provided by the present invention may occur across share path names, server names and file system or data access protocols. Thus, this approach may be used to consolidate storage, for example, on a Microsoft Windows™ Sharepoint server. As a result, the system and method provide significant advantages and benefits needed in contemporary computing.

While the invention is susceptible to various modifications and alternative constructions, certain illustrated embodiments thereof are shown in the drawings and have been described above in detail. It should be understood, however, that there is no intention to limit the invention to the specific forms disclosed, but on the contrary, the intention is to cover all modifications, alternative constructions, and equivalents falling within the spirit and scope of the invention.